

Sveučilište u Zagrebu  
Filozofski fakultet  
Odsjek za informacijske i komunikacijske znanosti  
Ak. god. 2011/2012.

**Matija Marohnić**

# Suvremene tehnologije za izradu web aplikacija

Završni rad

**Mentor:** dr. sc. Kristina Vučković

**Komentor:** dr. sc. Željko Agić

Zagreb, 2012.

# Sadržaj

<b>1</b>	<b>Uvod</b>	<b>3</b>
<b>2</b>	<b>Terminologija</b>	<b>4</b>
<b>3</b>	<b>Web preglednici</b>	<b>6</b>
3.1	Internet Explorer . . . . .	6
3.2	Google Chrome . . . . .	7
<b>4</b>	<b>Tehnologija</b>	<b>8</b>
4.1	HTML5 . . . . .	8
4.2	CSS3 . . . . .	9
4.3	JavaScript . . . . .	10
4.4	Ruby . . . . .	10
4.4.1	Rails . . . . .	11
4.4.2	bcrypt . . . . .	12
4.4.3	Paperclip . . . . .	12
4.4.4	RSpec . . . . .	12
4.4.5	Haml . . . . .	13
4.4.6	Sass . . . . .	13
4.4.7	Coffee-Script . . . . .	15
4.5	PostgreSQL . . . . .	15
4.6	Typekit . . . . .	15
4.7	IcoMoon . . . . .	16
4.8	Dropbox . . . . .	16
4.9	Heroku . . . . .	16
4.10	Git . . . . .	17
<b>5</b>	<b>Zaključak</b>	<b>18</b>
<b>6</b>	<b>Bibliografija</b>	<b>19</b>

# 1 Uvod

U vrijeme pisanja ovog rada već dva mjeseca razvijam aplikaciju za koju se nadam da će potaknuti djecu da više čitaju lektiru.

Postoje već slične aplikacije koje omogućavaju igranje kvizova, ali su velike, teške za implementirati, a često i skupe. Ali to je s razlogom! Kviz je prilično kompleksan koncept za realizirati na jednostavan i fleksibilan način. Ova aplikacija ne pokušava donijeti univerzalno rješenje, već specijalizirano.

Cilj ove aplikacije je napraviti usvajanje znanja iz lektira zabavnim, uzbudljivim i interaktivnim procesom. Nadam se da će mi, uz pomoć škola koja će testiraju moju aplikaciju, to i uspjeti.

U ovome ću se radu koncentrirati na tehnologiju koju sam koristio u razvijanju aplikacije, ukratko ću obrazložiti zašto sam donio određene odluke, koje su alternative alatima koje sam izabrao i sl. Implementaciju ću opisati u slijedećem radu koji će se natjecati za rektorovu nagradu.

U drugom ću se poglavlju kratko osvrnuti na terminologiju koju koristim u ovom radu. U trećem ću poglavlju ukratko opisati dva web preglednika, Internet Explorer i Google Chrome, te ću obrazložiti koji koristim i zašto. U četvrtom poglavlju opisat ću niz tehnologija koje sam koristio u izradi aplikacije. Na kraju rada dati ću i kratak zaključak o trenutnom stanju tehnologija za izradu web aplikacija.

## 2 Terminologija

Prije nego što počnem s detaljima, potrebno je razjasniti višeznačnu terminologiju razvoja mrežnih stranica. S obzirom da je to područje koje jako brzo napreduje, pojmovi koji slijede nemaju čvrstih definicija, ali ovo je jedna od interpretacija koju ću koristiti u ovom radu.

**Web stranica i web aplikacija.** Web stranicom se smatra sadržaj na internetu koji je statičan u smislu da uglavnom ne možemo utjecati na njezin sadržaj. Takve stranice služe za prezentacijske svrhe. Web aplikacija je, s druge strane, popularan termin koji opisuje web stranicu koja se ponaša kao što bi se ponašala aplikacija na našem računalu, samo ima više mogućnosti jer je automatski povezana na internet. Možemo utjecati na njen sadržaj i djelovati na ostale korisnike te aplikacije (ukoliko postoje).

**Korisničko iskustvo** (eng. *user experience*). Apstraktan termin koji opisuje kako se korisnik *osjeća* dok koristi web stranicu. Je li stranica intuitivna za korištenje, ponašaju li se njeni elementi na očekivan način i sl.

**Client-side i server-side.** Prvi se termin odnosi na klijenta, odnosno web preglednika, dok se drugi odnosi na poslužitelja<sup>1</sup>.

**Front-end i back-end.** Prvi se termin odnosi na ono što posjetitelj vidi kad posjeti web stranicu, a drugi se termin uglavnom odnosi na administracijsko sučelje.

**Web dizajner i web developer.** Iako točan opseg njihovog polja nije definiran, web dizajnerom se obično smatra osoba koja je specijalizirana za izgled stranice i korisničko iskustvo, a web developer je osoba koja se bavi programerskim dijelom web stranice, obično vezanim za poslužitelje. Web dizajnere i web developere ponekad kraće zovem *dizajnerima* i *developerima*.

**Hosting.** Usluga koju pruža web poslužitelj<sup>2</sup>.

**Backup.** Spremanje kopija datoteka na drugo mjesto, tako da u slučaju gubitka „originala“ datoteke nisu izgubljene.

**Open-source program.** Besplatan program otvorenog kôda, što znači da svatko može vidjeti kôd i po potrebi ga izmjenjivati i popravljati.

---

<sup>1</sup>Npr. kada korisnik klikne na poveznicu (eng. *link*), njegov preglednik (klijent) zatraži stranicu od udaljenog računala (poslužitelja) na kojemu sa nalazi web stranica, koji ju zatim šalje natrag pregledniku.

<sup>2</sup>Osnovna usluga koju pruža svaki web poslužitelj je sposobnost dostavljanja željene web stranice putem određene domene. Poslužitelji mogu imati i napredne funkcionalnosti, kao što je postavljanje e-mail adrese, podrška za druge jezike osim PHP-a, udaljeni pristup i sl.

**Pseudo-nasumičnost.** Programi nemaju sposobnost biranja potpuno nasumičnog broja, pa do njega dolaze kompliciranim algoritmom, a broj koji je nastao pomoću nekog algoritma ne može u potpunosti biti nasumičan, zato se kaže da je on pseudo-nasumičan. U ovom radu „pseudo-nasumičnost“ ću skratiti na „nasumičnost“.

**Kompajliranje.** Proces u kojem se kôd napisan u jeziku više razine prevađa u jezik niže razine kako bi se mogao izvršavati.

**Font.** Skup znakova dizajniranih u istom stilu koji se mogu koristiti za elektroničko pisanje.

**Biblioteka** (eng. *library*). Skup varijabla i funkcija koji pomaže u korištenju određenog (programskog) jezika.

**VCS (Version Control System).** Sustav za organizirano spremanje promjena u aplikaciji. Obično se odabere skup promjena koje su zajedno postigle određeni cilj, spremi ih se i napiše se opis promjene. Mape (eng. *folder*) koje se na taj način razvijaju zovu se *repozitoriji*.

Web stranicu ponekad kraće zovem *stranicom*, a web preglednik *preglednikom*.

Pojmove koje nisam naveo nemaju još odgovarajući hrvatski prijevod, stoga ću ih jednostavno pisati u izvornom obliku, u *kurzivu*.

## 3 Web preglednici

Moja aplikacija je trenutno optimizirana za Google Chrome, a nije dobro podržana na Internet Explorer-u. Upravo je to razlog zašto ću na ovom mjestu ukratko opisati samo ta dva preglednika iako ih na tržištu ima više, kao što su Mozilla Firefox, Apple Safari i Opera.

### 3.1 Internet Explorer

S obzirom da Internet Explorer web preglednik koji još uvijek zauzima velik postotak tržišta, želim obrazložiti zašto ga trenutno ne podržavam. Naime, prema mom mišljenju, Microsoft Internet Explorer je najgori od svih web preglednika. Postoji mnogo razloga, ovo je samo nekoliko:

- **Ažurnost** – Internet Explorer kaska za gotovo svakim standardom koje je njegova konkurencija odavno implementirala. To nije problem samo njegovim korisnicima, nego i developerima i dizajnerima, jer su ga u većini slučajeva prisiljeni podržavati.
- **IE6** – U vrijeme pisanja ovog rada 6% svijeta još uvijek koristi zastarjeli Internet Explorer 6 (koji je većinom zastupljen u Kini), a to se dogodilo jer je njegov razvoj stagnirao na nekoliko godina kada nije bilo konkurencije.
- **Razvijanje web stranica** – Iznimno je teško otkriti zašto nešto ne radi u Internet Exploreru, a to je zbog vrlo lošeg mehanizma detektiranja pogrešaka. Također, ima vrlo spor algoritam za učitavanje stranica, što znatno produljuje postupak testiranja.
- **Sigurnost** – Internet Explorer ima veći broj sigurnosnih propusta od svih preglednika.
- **JavaScript** – Podržanost JavaScript-a na Internet Explorer-u vrlo je loša, zbog čega je jQuery otprilike duplo veći od veličine koju je mogao zauzimati.
- **Zatvoren** – Internet Explorer je preglednik zatvorenog koda, za razliku od drugih preglednika kao što su Google Chrome i Mozilla Firefox.
- **Podržanost** – Internet Explorer je dostupan samo za Windows, dok je njegova konkurencija dostupna za gotovo svaki operacijski sustav. To je glavni nedostatak koji mi otežava testiranje aplikacije, jer za razvijanje koristim Ubuntu (jedna od distribucija Linux-a), a Internet Explorer nije podržan na tom sustavu.<sup>[1]</sup>

## 3.2 Google Chrome

Aplikaciju sam razvijao u Google Chrome-u, stoga na njemu najbolje i radi. Postoji mnogo razloga zašto sam odabrao baš taj preglednik:

- **WebKit** – Google Chrome je, kao i Safari, temeljen na **WebKit**-u, moćnom *open-source* sustavu za prikazivanje stranica, kojeg razvijaju razni programeri iz poznatih tvrtki kao što su Apple Inc., Nokia, RIM i Samsung. WebKit trenutno obuhvaća većinu tržišta i, osim na Google Chrom-u i Safari-u, koristi se i na mnogim mobilnim operacijskim sustavima, kao što su BlackBerry, iOS, Tablet OS itd.<sup>[3]</sup>
- **Ažurnost** – Google Chrome podržava velik dio novih HTML5 i CSS3 mogućnosti.
- **Brzina** – Google Chrome je najbrži preglednik od svih koje sam koristio.
- **Omnibar** – Google Chrome je povezao polje za URL i polje za Google Search u jedno polje pod nazivom **Omnibar**, što ga čini iznimno jednostavnim i lakim za korištenje.
- **Otvorenost** – Google je izdvojio većinu koda kojeg Google Chrome koristi i otvorio ga za javnost pod nazivom **Chromium**.
- **Podržanost** – Google Chrome je dostupan na gotovo svim operacijskim sustavima, a za one na kojima nije, dostupan je Chromium.
- **Web inspektor** – Google Chrome, kao i Safari, ima ugrađen alat koji pomaže u razvijanju stranica. Sastoji se od mnogo komponenata, ima konzolu koja javlja ako se dogodila greška na stranici, prikaz koliko je vremena trebalo da se učita određena datoteka na stranici, daje savjete kako optimizirati brzinu stranice, omogućava jednostavno testiranje JavaScript-a itd.<sup>[2]</sup>

Preglednici imaju mnogo više karakteristika od ovih koje sam naveo, a istaknuo sam iste jer su mi izrazito bitne pri izradi web stranica i zbog njih je Google Chrome moj prvi izbor preglednika.

## 4 Tehnologija

Kod razvijanja web stranica izbor tehnologije je prilično ograničen. Jezici koje klijent koristi su HTML, CSS i JavaScript, oni nemaju alternativa i prisiljeni smo ih koristiti na jedan način ili na drugi. Međutim, poslužitelj je računalo, dakle mogućnosti su puno veće. Osim klijenta i poslužitelja treba obratiti pažnju na rukovanje promjenama u aplikaciji, pa bi prema mom mišljenju svaki ozbiljniji developer trebao koristiti VCS kako bi si olakšao izradu aplikacije, jer ga VCS potiče da razmišlja na modularan način, olakšava mu pregledavanje promjena, pomaže mu u dokumentaciji itd. U svom radu sam koristio slijedeće tehnologije: HTML5, CSS3, JavaScript, Ruby, PostgreSQL, Typekit, IcoMoon, GitHub, Dropbox, Heroku i Git.

**Klijent.** Za *client-side* sam koristio HTML5, CSS3 i JavaScript.

**Poslužitelj.** Za *server-side* sam koristio programski jezik Ruby i PostgreSQL bazu podataka.

**Vanjski servisi.** Od vanjskih servisa koristio sam Typekit za fontove, IcoMoon za ikone, GitHub za *backup* repozitorija, Dropbox za *upload* datoteka i Heroku za *hosting*.

**VCS.** Za sustavno rukovanje promjenama koristio sam Git.

### 4.1 HTML5

HTML (**H**yper**T**ext **M**arkup **L**anguage) je jezik koji opisuje strukturu stranice. HTML specifikacije održava World Wide Web Konzorcij.<sup>[4]</sup> Svaki web preglednik ima ugrađene stilove koji opisuju kako će određeni HTML elementi izgledati, ali sâm HTML ne određuje prikaz svojih elemenata. Vrlo je bitno usvojiti taj koncept jer razdvaja strukturu web stranice od njezinog izgleda.

HTML ima mnogo verzija, HTML 3.0, HTML 4.0, HTML 4.01, XHTML 1.1, HTML5 itd. Kako bi web preglednik znao po kojim da pravilima interpretira web stranicu, HTML verzija se deklarira na samom vrhu HTML dokumenta. Npr. HTML5 dokument deklarira se ovako:

```
<DOCTYPE! html>
```



U HTML-u je neko vrijeme bio popularan XHTML (EXtensible HyperText Markup Language). To je ideja inspirirana XML-om, a glavne značajke su stroga sintaksa i mogućnost definiranja novih elemenata. Međutim, budući da ga Internet Explorer nije podržavao, XHTML je bio samo ograničavajuć, jer se nisu mogle koristiti njegove prednosti. Grupa koja je razvijala XHTML imala je planove za XHTML 2.0, ali njihov je rad polako utihnuo i počeo se razvijati HTML5.

HTML5 je živući standard (eng. *living standard*), što znači da nikada neće biti gotov i da će HTML5 uvijek pokazivati na najnoviju verziju HTML-a.<sup>[5]</sup> HTML5 je po mnogočemu suprotan XHTML-u, ponajviše s obzirom na sintaksu. HTML5 je iznimno liberalan, što je osvježavajuće nakon stroge sintakse XHTML-a. Toliko je liberalan da je svaki validan XHTML dokument ujedno i validan HTML5 dokument.

HTML5 donosi specifikacije za mnoge funkcionalnosti koje su web developeri oduvijek htjeli, ali nisu imali odgovarajuću podršku te su ih emulirali drugim tehnologijama, kao što su Flash i Java. Jedan od primjera je podrška za video i audio za koju se prije uvijek koristio Flash, a sada se može koristiti u svim modernim web preglednicima bez posrednika.<sup>[6]</sup>

Koje su prednosti toga da preglednici podržavaju nove funkcionalnosti bez pomoći druge tehnologije? Jedna od prednosti je to što korisnik dobiva veću kontrolu nad sadržajem. Npr. postoji jedan HTML5 atribut koji se može pridružiti tekstualnom polju i učiniti ga da se automatski fokusira kada se stranica učita (kao npr. kod Google pretraživanja), a ako korisnik to ne želi, moći će tu funkcionalnost isključiti u svom pregledniku (jednom kada preglednik to odluči omogućiti). Kada se ta funkcionalnost emulirala JavaScript-om, nije bilo moguće to isključiti, a da se ne isključi čitavi JavaScript, što je iritiralo mnoge korisnike.

## 4.2 CSS3

Za razliku od HTML-a, koji opisuje strukturu stranice, CSS (Cascading Style Sheets) opisuje **izgled** stranice. Nastao je kao rješenje koje razdvaja prezentaciju od strukture stranice, jer se izgled web stranice prije definirao prezentacijskim HTML elementima, što je brzo postalo teško za održavati.

CSS omogućava dizajneru da na drugačiji način prikazuje stranicu kada je na ekranu, kada se ispisuje, kada ju čita pomoćni program za slijepe osobe (eng. *screen reader*) itd. Također, za razliku od prezentacijskog HTML-a, isti se CSS dokument može učitati na više različitih stranica, što puno olakšava izmjenu.

Za razliku od prijašnjih verzija, CSS3 je koncipiran kao skup modula koji donose određene funkcionalnosti kao što su zaobljeni rubovi, selektori razine 3, animacije, gradijenti i sl. Za takve efekte su dizajneri prije trebali koristiti JavaScript i slike.

HTML5 je živući standard, ali CSS3 nije. WWW Konzorcij je već počeo raditi na CSS4 standardu koji će podržavati naprednije selektore i mnoge druge mogućnosti.<sup>[7]</sup>

## 4.3 JavaScript

JavaScript je skriptni jezik koji podržava objekto-orijentirani, imperativni i funkcionalni stil programiranja. U zadnjih nekoliko godina je vrlo popularan jer je dovoljno sazrio za ozbiljnu upotrebu.

JavaScript čini mnoge procese bržim jer se preglednik ne mora osvježavati, a neke se funkcionalnosti niti ne mogu izvesti bez JavaScript-a, kao što je npr. *drag & drop* tehnika. Međutim, korištenje JavaScript-a zahtijeva odgovornost – većina stranice treba biti upotrebljiva i bez JavaScript-a.

Mnogi ljudi povezuju, a ponekad i zamjenjuju JavaScript s **Javom**. JavaScript je skriptni jezik za web okruženje, dok je Java programski jezik. Osim sličnosti u pravilima imenovanja, ta dva jezika su potpuno nepovezana i služe različitoj svrsi.<sup>[8]</sup>

Pristupanje elementima na stranici pomoću čistog JavaScript-a je prilično teško i neugodno iskustvo, zbog nekonzistentne implementacije u različitim preglednicima.<sup>[9]</sup> Zato koristim jQuery, JavaScript biblioteku koja popravljala i nadopunjava JavaScript. Npr. dobavljanje elemenata koji imaju određenu klasu je nešto što developeri često rade. Međutim, JavaScript to ne podržava (barem ne na svim preglednicima). jQuery olakšava pristupanje elementima s kratkom i konzistentnom CSS-olikom sintaksom:<sup>[10]</sup>

```
$("#main"); // dobavlja element na stranici koji ima ID "main"
$(".post"); // dobavlja sve elemente koji imaju klasu "post"
```

Postoje nekoliko sličnih JavaScript biblioteka kao što su Prototype, YUI i Zepto, ali izabrao sam **jQuery** zbog njegove rasprostranjenosti i podrške. Međutim, kasnije namjeravam koristiti Zepto umjesto jQuery-a jer je puno elegantniji i zauzima manje prostora.

## 4.4 Ruby

*Ruby is a dynamic, open source programming language with a focus on simplicity and productivity. It has an elegant syntax that is natural to read and easy to write. —[ruby-lang.org](http://ruby-lang.org)<sup>[11]</sup>*

Ruby je programski jezik kojeg je 1995. razvio Yukihiro Matsumoto, i od tada je privlačio sve više ljudi sve dok 2006. nije postao svjetski prihvaćen.

Ruby je objektno-orijentiran programski jezik. Njegova ključna karakteristika, po kojoj se razlikuje od ostalih objektno-orijentiranih jezika, je što je u Ruby-u **sve** objekt. Dakle, čak i kod jednostavnih aritmetičkih radnji, kao npr. `1 + 1`, ono što se zapravo događa je da broj 1 poziva metodu `+` s argumentom 1. Prethodna aritmetička radnja može se zapisati i ovako `1.+(1)`.

Kod objektno-orijentiranih jezika jedan od glavnih problema je nasljeđivanje. Većina jezika ili dopušta nasljeđivanje od samo jedne klase, pa imaju problem limitiranosti, ili dopušta nasljeđivanje od više klasa, što je pak previše kompleksno. Ruby, s druge strane, dopušta nasljeđivanje od samo jedne klase, ali nudi mogućnost uključivanja skupova metoda u klasu.

Ruby ima vrlo jednostavnu vizualnu reprezentaciju raznih vrsta varijabli:

- `var` može biti lokalna varijabla
- `@var` je članska varijabla
- `@@var` je klasna varijabla
- `$var` je globalna varijabla.

Ruby je vrlo dinamičan jezik. Toliko je dinamičan da dopušta izmjenu, dodavanje, i brisanje elemenata iz standardne biblioteke. Recimo da programer za zbrajanje umjesto simbola `+` želi koristiti riječ `plus`. Potrebno je samo dodati tu metodu u klasu brojeva:

```
class Numeric
  def plus(x)
    self.+(x)
  end
end
```

```
y = 5.plus 6
```

Postoji preko 40,000 besplatnih Ruby biblioteka (eng. *gem*) dostupnih za korištenje. Od biblioteka koje sam koristio u ovoj aplikaciji navest ću najvažnije.

### 4.4.1 Rails

*Ruby on Rails is an open-source web framework that's optimized for programmer's happiness and sustainable productivity. It lets you write beautiful code by favoring convention over configuration. —[rubyonrails.org](http://rubyonrails.org)<sup>[12]</sup>*

Rails je 2003. godine razvio David Heinemeier Hansson (poznat po inicijalima DHH). To je veliki projekt kojeg razvija više od 2,100 programera diljem svijeta.

Rails je „samouvjeren“ alat. On radi pretpostavku da postoji najbolji način za organizirati kôd, i dizajniran je da ga potiče – a u nekim slučajevima i da obeshrabruje alternative. Filozofija Rails-a uključuje sljedeća načela:

- **DRY** (**D**on't **R**epeat **Y**ourself) – Loše je pisati isti kôd na više različitih mjesta u projektu.

- „*Convention Over Configuration*“ – Rails pretpostavlja što i kako korisnik želi postići određeni cilj, a ne zahtjeva od korisnika da specificira svaki sitni detalj u konfiguraciji.
- **REST** (**RE**presentational **S**tate **T**ransfer) – Organizacija aplikacije s naglaskom na resurse i standardne HTTP „glagole“ (GET, POST, PUT, PATCH, DELETE) je najbrži način za razvitak stabilne aplikacije.

Rails koristi tzv. **MVC** (**M**odels-**V**iews-**C**ontrollers) arhitekturu, koja se sastoji od sloja za čuvanje i manipuliranje podacima (baza podataka), sloja za dobavljanje tih podataka, i sloja za prezentiranje tih podataka, najčešće u obliku HTML stranice.

Mnoge poznate web aplikacije napravljene su u Rails-u, npr. [Twitter](#) i [GitHub](#).

#### 4.4.2 bcrypt

bcrypt je funkcija za kriptiranje osjetljivih podataka. Bolja je u odnosu na poznate (npr. MD5) jer je algoritam za kriptiranje namjerno napisan da bude spor. U usporedbi s MD5, bcrypt je 300 puta sporiji, što je korisno kada hakeri pokušavaju probiti lozinku jer trebaju kriptirati velik broj potencijalnih lozinki. S druge strane, vrijeme potrebno da se kriptira jedna lozinka (npr. kod registracije) je zanemarivo dulje, tako da korisnik niti ne primjećuje zadržku.

#### 4.4.3 Paperclip

Paperclip je biblioteka za jednostavno spremanje datoteka koje su se učitale preko web forme na poslužitelj. Nudi razne validacije, kao što su prisutnost i veličina.<sup>[13]</sup>

Paperclip također nudi i spremanje slike u različitim oblicima. Npr. često je potrebno da sve slike u nekom kontekstu budu istih dimenzija, pa je moguće namjestiti da se pri učitavanju uz original uvijek sprema i verzija te slike u potrebnim dimenzijama. Pojam „oblika“ slike nije ograničen samo na dimenzije, naime, slika se može na razne načine obraditi, poput npr. dodavanja crno-bijelog efekta.

#### 4.4.4 RSpec

RSpec je moćno sučelje za testiranje Ruby koda i služi kao alternativa standardnom sučelju za testiranje koje dolazi uz Ruby. RSpec omogućava pisanje nevjerojatno čitljivih testova, poput:<sup>[14]</sup>

```
post.title.should_not be_empty
post.views_count.should be_greater_than(10)
```

## 4.4.5 Haml

Haml (XHTML abstraction markup language) je jezik koji apstrahira HTML, daje mu funkcije i čini ga bržim za pisanje i čitanje.<sup>[15]</sup>

Nastao je u vrijeme kada je XHTML bio popularan, jer tada nije bilo HTML5 semantičkih elemenata, pa se često koristio element `<div>`. Ovako bi izgledao dio koda izvađen iz prosječnog XHTML dokumenta:

```
<div id="wrapper">
  <div id="main">
    <div class="post">
      <div class="header">
        <h1 class="title">
          A post
        </h1>
      <div class="date">
        January 1st, 2012.
      </div>
    </div>
    <div class="footer">
      Author info.
    </div>
  </div>
</div>
```

Ovakav kôd brzo postaje kompliciran za čitati i izmjenjivati. Haml ga može pojednostaviti jednostavnijom sintaksom, inspiriranom CSS-om:

```
#wrapper
  #main
    .post
      .header
        h1.title A post
      .date January 1st, 2012.
      .footer Author info.
```

## 4.4.6 Sass

Sass (Syntactically awesome style sheets) je CSS preprocesor koji opskrbljuje CSS dodatnim mogućnostima kao što su funkcije, varijable i sl. Podržava dvije sintakse, prva je nalik normalnom CSS-u, a druga je indentirana, inspirirana Haml-om. Prva sintaksa je popularnija jer je lakša za naučiti i fleksibilnija je.<sup>[16]</sup>

Što CSS kôd postaje kompliciraniji, to je teže održavati ga jer je vrlo redundantan, pogotovo kada se koriste CSS3 efekti. Uzmimo za primjer ovaj CSS3 kôd:

```
#wrapper {
  -webkit-border-radius: 3px;
  -moz-border-radius: 3px;
  -ms-border-radius: 3px;
  -o-border-radius: 3px;
  border-radius: 3px;
}

#wrapper .post {
  -webkit-border-radius: 6px;
  -moz-border-radius: 6px;
  -ms-border-radius: 6px;
  -o-border-radius: 6px;
  border-radius: 6px;
}

#wrapper .post .time {
  -webkit-border-radius: 12px;
  -moz-border-radius: 12px;
  -ms-border-radius: 12px;
  -o-border-radius: 12px;
  border-radius: 12px;
}
```

Osim očitog ponavljanja `border-radius` svojstva primjetite i ponavljanje selektora. Ovaj se kôd može pojednostaviti u Sass-u:

```
@mixin border-radius($radius) {
  @each $vendor-prefix in webkit, moz, ms, o {
    -#{$vendor-prefix}-border-radius: $radius;
  }
  border-radius: $radius;
}

#wrapper {
  @include border-radius(3px);

  .post {
    @include border-radius(6px);

    .date { @include border-radius(12px);
  }
}
```

## 4.4.7 Coffee-Script

CoffeeScript je jezik koji se kompajlira u JavaScript, a sintaksa mu je jednostavnija i elegantnija od JavaScript-a, više je nalik Ruby-u.

JavaScript ima mnogo specifičnih slučajeva u kojima se naredbe ne izvršavaju na očekivani način, pa su takve situacije prilično nezgodne za početnike. Problem je u tome što te specifičnosti ne postaju logične nakon puno iskustva, nego se jednostavno trebaju prihvatiti kao dio jezika. Npr. ako se varijabla u JavaScript-u deklarira bez ključne riječi `var`, interpreter pretpostavi da je developer htio deklarirati globalnu varijablu (a ne lokalnu), što je najgora moguća pretpotavka. Takvi problemi mogu biti izrazito teški za otkriti nekome tko nije upoznat s takvim specifičnim slučajevima.

Cilj CoffeeScript-a je osloboditi developera tih obaveza i omogućiti mu da se koncentrira na logiku koda, a ne na sintaksu. Osim popravljanja JavaScript-ovih „grešaka“, CoffeeScript raspolaže i raznim funkcionalnostima koje JavaScript nema.<sup>[17]</sup> Npr. obratimo pozornost na ovaj JavaScript kôd:

```
names = ['Andy', 'Mike', 'Susy', 'Craig'];
for (var i=0; i<names.length; i++) {
  alert(names[i]);
}
```

Ovaj kôd prolazi kroz listu imena i ispisuje svaki element. Za postizanje tog cilja trebali smo deklarirati varijablu koja će služiti kao brojač, zatim nakon svake iteracije provjeravati je li taj brojač manji od duljine liste i na kraju svake iteracije povećati brojač za 1. U CoffeeScript-u bi se to moglo napisati na jednostavniji način:

```
alert(name) for name in ['Andy', 'Mike', 'Susy', 'Craig']
```

## 4.5 PostgreSQL

PostgreSQL je vrlo moćan open-source **DBMS** (**D**ata**B**ase **M**anagement **S**ystem). Nastao je prije 15 godina, i danas je još uvijek u aktivnom razvoju. Ima puno više mogućnosti od standardnih DBMS-ova (poput MySQL-a); od raznih tipova, kao što su npr. `point` (točka) i `hstore` (ključ-vrijednost), pa sve do vrlo naprednih mogućnosti kao što je pretraživanje tekstova uz korištenje raznih specijaliziranih indeksa i riječnika.

## 4.6 Typekit

Jedna od bitnijih novosti u web dizajnu je mogućnost korištenja novih fontova. Prije su dizajneri bili ograničeni samo na fontove za koje se smatra da svi više-manje imaju.

Typekit je profesionalni servis koji pruža upravo takvu uslugu. Ima veliki izbor kvalitetnih fontova na korištenje, koji se učitavaju na web stranicu putem JavaScript-a. Galerija fontova omogućava traženje fontova po različitim kriterijima, kao što su tip fonta (serif, sans-serif, monospace...), debljina, kontrast, visina slova i sl.<sup>[18]</sup>

Postoji više od 10 drugih servisa kao što su WebINK, Typotheque i Google Fonts koji također nude sličnu uslugu, ali odabrao sam Typekit zato što ima bolji financijski plan, više kvalitetnih fontova na izbor, jednostavniji je i stabilniji i ima izvrsnu korisničku podršku.<sup>[20]</sup>

## 4.7 IcoMoon

Zbog navedene dobre podrške za fontove počeli su se razvijati i fontovi koji kao znakove koriste ikone. Takvi fontovi idealni su za upotrebu na web stranicama jer su onda ikone vektorske, dakle izgledaju oštro na bilo kojoj rezoluciji.

IcoMoon je servis koji, između ostaloga, omogućava slaganje vlastitih fontova. Moguće je izmjenjivati već postojeći font ili napraviti novi od vektorskih slika. Glavna prednost korištenja IcoMoon servisa je mala veličina datoteka, jer se u njima nalaze samo ikonice koje su potrebne, a ne cijeli set.<sup>[19]</sup>

## 4.8 Dropbox

Dropbox je besplatna web aplikacija koja omogućuje spremanje, sinkronizaciju i dijeljenje datoteka preko interneta. Osnovani su ga je 2007. Drew Houston i Arash Ferdowsi, dvojica MIT studenata kojima je dosadilo koristiti e-mail za prijenos datoteka. Danas više od 50 milijuna ljudi koristi Dropbox za dijeljenje slika, dokumenata i video zapisa. Talentirane se osobe mogu pridružiti daljnjem razvoju aplikacije.<sup>[21]</sup>

Dropbox podržava udaljenu interakciju, pa je koristan za spremanje datoteka iz aplikacije.

## 4.9 Heroku

Heroku je servis za *hosting* web aplikacija. Razlikuje se od drugih *hosting* servisa po tome što već ima sve potrebne konfiguracije, kao npr. podrška za Git, čineći time cijeli proces potpuno bezbolnim.<sup>[22]</sup>



## 4.10 Git

*Git is a free and open source distributed version control system designed to handle everything from small to very large projects with speed and efficiency.*  
—[git-scm.com](https://git-scm.com)<sup>[23]</sup>

Git je sustav za rukovanje promjenama. Elegantan je, zauzima malo prostora, lagan je za naučiti, siguran je i iznimno moćan.

Postoji više sličnih sustava koji su prije bili popularni, kao SVN, ali izabrao sam Git jer ima mnogo prednosti, a nisam naišao niti na jedan nedostatak. Također, trenutni centar open-source svijeta, GitHub, je servis za Git repozitorije, stoga smatram da svaki ozbiljniji *open-source* programer koristi Git.

Najpoznatiji servis za Git repozitorije je GitHub, koji je praktički sinonim je za *open-source*. Svi se važniji *open-source* projekti nalaze na GitHub-u, to je savršeno mjesto za surađivanje s drugim programerima, upravo zbog jednostavnosti Git-a. Jedna pozitivna nuspojava surađivanja preko GitHub-a je ta što uvijek postoji kopija repozitorija na internetu, dakle to je automatski *backup*.

## 5 Zaključak

Vjerojatno ste dosada primjetili koliko postavke za jednostavnu web aplikaciju mogu biti kompleksne, ali ako pogledamo samo 10-ak godina unatrag, možemo primjetiti da mnoge navedene tehnologije nisu niti postojale. Svijet tehnologije se jako brzo mijenja, nakon višegodišnjeg iskustva u razvoju web stranica, još uvijek se ponekad zateknem kako više učim o novim tehnologijama nego što obavljam posla. Za postizanje ravnoteže između učenja i rada potrebno je puno truda, jer se može lako dogoditi da se baš nakon što ste usvojili jedan dio tehnologije pojavi nova tehnologija za koju se ispostavi da je bolja od one koju ste vi naučili.

Potrebe korisnika se eksponencijalno povećavaju i uloga dizajnera i developera je dati im ono što traže. Npr. JavaScript se prije jako oprezno koristio jer je bio nezreo za upotrebu, a danas je JavaScript nužna tehnologija za stvaranje ugodnog korisničkog iskustva.

Aplikacija je *open-source*, njezin se kôd može [vidjeti](#) na GitHub-u i tamo se mogu vidjeti detalji implementacije navedenih tehnologija.

## 6 Bibliografija

- [1] Internet Explorer // Wikipedia: The Free Encyclopedia. 14.9.2012. URL: [http://en.wikipedia.org/wiki/Internet\\_Explorer](http://en.wikipedia.org/wiki/Internet_Explorer) (21.9.2012.)
- [2] Google Chrome // Wikipedia: The Free Encyclopedia. 21.9.2012. URL: [http://en.wikipedia.org/wiki/Google\\_Chrome](http://en.wikipedia.org/wiki/Google_Chrome) (21.9.2012.)
- [3] WebKit // Wikipedia: The Free Encyclopedia. 30.8.2012. URL: <http://en.wikipedia.org/wiki/WebKit> (21.9.2012.)
- [4] HTML // Wikipedia: The Free Encyclopedia. 19.9.2012. URL: <http://en.wikipedia.org/wiki/HTML> (21.9.2012.)
- [5] Mozilla Development Network : HTML. URL: <https://developer.mozilla.org/en-US/docs/HTML> (21.9.2012.)
- [6] HTML5 Rocks : a resource for open web HTML5 developers. URL: <http://www.html5rocks.com/en/> (18.9.2012.)
- [7] Cascading Style Sheets // Wikipedia: The Free Encyclopedia. 20.9.2012. URL: <http://en.wikipedia.org/wiki/Css> (21.9.2012.)
- [8] JavaScript // Wikipedia: The Free Encyclopedia. 21.9.2012. URL: <http://en.wikipedia.org/wiki/Javascript> (21.9.2012.)
- [9] Mozilla Development Network : JavaScript. URL: <https://developer.mozilla.org/en-US/docs/JavaScript> (21.9.2012.)
- [10] jQuery JavaScript Library. URL: [http://docs.jquery.com/Main\\_Page](http://docs.jquery.com/Main_Page) (18.9.2012.)
- [11] Ruby Programming Language. URL: <http://www.ruby-lang.org/en/> (18.9.2012.)
- [12] Ruby on Rails. URL: <http://rubyonrails.org> (18.9.2012.)
- [13] Paperclip. 11.9.2012. URL: <https://github.com/thoughtbot/paperclip/blob/master/README.md> (21.9.2012.)
- [14] RSpec Expectations. URL: <https://www.relishapp.com/rspec/rspec-expectations/v/2-11/docs/built-in-matchers/operator-matchers> (21.9.2012.)

- [15] Haml : markup haiku. URL: <http://haml.info/> (20.9.2012.)
- [16] Sass : style with attitude. URL: <http://sass-lang.com/> (20.9.2012.)
- [17] CoffeeScript. URL: <http://coffeescript.org/> (20.9.2012.)
- [18] Typekit: help. URL: <https://typekit.com/help> (18.9.2012.)
- [19] IcoMoon. URL: <http://icomoon.io/#toDocs> (18.9.2012.)
- [20] Review of Popular Web Font Embedding Services. 20.10.2010.  
URL: <http://www.smashingmagazine.com/2010/10/20/review-of-popular-web-font-embedding-services/> (20.9.2012.)
- [21] About Dropbox. URL: <https://www.dropbox.com/about> (20.9.2012.)
- [22] Heroku : cloud application platform. URL: <http://www.heroku.com> (18.9.2012.)
- [23] Git. URL: <http://git-scm.com> (18.9.2012.)